

Contents

Formats of Files Used by Networking Commands (SFF)

intro	introduction to files
aliases	aliases file for sendmail
hosts	list of hosts on network
hosts.equiv	list of trusted hosts
inetd	configuration file for inetd
localhosts	configuration file for sendmail
netrc	login file for remote networks
networks	names and numbers for the Internet
protocols	list of Internet protocols
resolver	resolver configuration file
rhosts	remote equivalent users
sendmail.cf	configuration file for sendmail
services	list of Internet services
sockcf	socket configuration file
strcf	STREAMS configuration file for STREAMS TCP/IP
uucpindomain	configuration file for sendmail

August 1917

1. The first of the following is a list of the names of the persons who have been elected to the office of the President of the United States since 1789.

1789	George Washington	1801	James Madison
1793	Thomas Jefferson	1809	Andrew Jackson
1797	John Adams	1817	James Monroe
1803	Thomas Jefferson	1821	John Quincy Adams
1805	James Madison	1825	Andrew Jackson
1809	Andrew Jackson	1829	Andrew Jackson
1813	James Monroe	1833	Andrew Jackson
1817	James Monroe	1837	Andrew Jackson
1821	John Quincy Adams	1841	Andrew Jackson
1825	Andrew Jackson	1845	Andrew Jackson
1829	Andrew Jackson	1849	Andrew Jackson
1833	Andrew Jackson	1853	Andrew Jackson
1837	Andrew Jackson	1857	Andrew Jackson
1841	Andrew Jackson	1861	Andrew Jackson
1845	Andrew Jackson	1865	Andrew Jackson
1849	Andrew Jackson	1869	Andrew Jackson
1853	Andrew Jackson	1873	Andrew Jackson
1857	Andrew Jackson	1877	Andrew Jackson
1861	Andrew Jackson	1881	Andrew Jackson
1865	Andrew Jackson	1885	Andrew Jackson
1869	Andrew Jackson	1889	Andrew Jackson
1873	Andrew Jackson	1893	Andrew Jackson
1877	Andrew Jackson	1897	Andrew Jackson
1881	Andrew Jackson	1901	Andrew Jackson
1885	Andrew Jackson	1905	Andrew Jackson
1889	Andrew Jackson	1913	Andrew Jackson
1893	Andrew Jackson	1917	Andrew Jackson
1897	Andrew Jackson	1921	Andrew Jackson
1901	Andrew Jackson	1925	Andrew Jackson
1905	Andrew Jackson	1929	Andrew Jackson
1909	Andrew Jackson	1933	Andrew Jackson
1913	Andrew Jackson	1937	Andrew Jackson
1917	Andrew Jackson	1941	Andrew Jackson
1921	Andrew Jackson	1945	Andrew Jackson
1925	Andrew Jackson	1949	Andrew Jackson
1929	Andrew Jackson	1953	Andrew Jackson
1933	Andrew Jackson	1957	Andrew Jackson
1937	Andrew Jackson	1961	Andrew Jackson
1941	Andrew Jackson	1965	Andrew Jackson
1945	Andrew Jackson	1969	Andrew Jackson
1949	Andrew Jackson	1973	Andrew Jackson
1953	Andrew Jackson	1977	Andrew Jackson
1957	Andrew Jackson	1981	Andrew Jackson
1961	Andrew Jackson	1985	Andrew Jackson
1965	Andrew Jackson	1989	Andrew Jackson
1969	Andrew Jackson	1993	Andrew Jackson
1973	Andrew Jackson	1997	Andrew Jackson
1977	Andrew Jackson	2001	Andrew Jackson
1981	Andrew Jackson	2005	Andrew Jackson
1985	Andrew Jackson	2009	Andrew Jackson
1989	Andrew Jackson	2013	Andrew Jackson
1993	Andrew Jackson	2017	Andrew Jackson
1997	Andrew Jackson	2021	Andrew Jackson

intro

introduction to formats of files used by networking commands

Description

This section outlines the formats of various files. The C struct declarations for the file formats are given where applicable. Usually, these structures can be found in header files under the directories `/usr/include`, `/usr/include/net`, `/usr/include/netinet`, or `/usr/include/sys`.

References of the type named(ADMN) refer to entries found in Section ADMN of the TCPIP Network Administrator's Reference.

aliases

aliases file for sendmail

Syntax

`/usr/lib/aliases`

Description

This file describes user id aliases used by `/usr/lib/sendmail`. It is formatted as a series of lines of the form

name: name_1, name2, name_3, ...

The *name* is the name to alias, and the *name_n* are the aliases for that name. Lines beginning with white space are continuation lines. Lines beginning with '#' are comments.

Aliasing occurs only on local names. Loops can not occur, since no message will be sent to any person more than once.

After aliasing has been done, local and valid recipients who have a ".forward" file in their home directory have messages forwarded to the list of users defined in that file.

See Also

`sendmail(ADMN)`

hosts

list of hosts on network

Description

The file `/etc/hosts` is a list of hosts that share the network, including the local host. It is referred to by programs that need to translate between host names and DARPA Internet addresses when the name server is not being used [See *named(ADMN)*.] Each line in the file describes a single host on the network and consists of three fields separated by any number of blanks or tabs:

address name aliases ...

where

- | | |
|-------------------|--|
| <i>address</i> | is the DARPA Internet address. Unless another type of address is required by some host on the network, <i>address</i> should be a Class A address, which takes the form <i>net.node</i> , where <i>net</i> is the network number from <code>/etc/networks</code> (see <i>networks</i> (4)), that must be between 0 and 127; and <i>node</i> is a value which must be unique for each host and be between 0 and 16777215. |
| <i>name</i> | is the official name of the host. If the host is a computer system running UNIX, it must claim this host name by executing <i>hostname</i> (TC) when it is initializing itself. |
| <i>aliases...</i> | is a list of alternate names for the host. Aliases can be used in network commands in place of the official name. |

It is suggested that you specify the *hostname* and the *node name* [see *hostname*(TC) and *uname*(C)] as aliases for one another for each machine listed in the `/etc/hosts` file.

The routines which search this file ignore comments (portions of lines beginning with #) and blank lines.

An internet address can actually take one of four forms:

- | | |
|------------|--|
| <i>A</i> | <i>A</i> is a simple 32-bit integer. |
| <i>A.B</i> | <i>A</i> is an eight-bit quantity occupying the high-order byte and <i>B</i> is a 24-bit quantity occupying the remaining bytes. This form is suitable for a Class A address of the form <i>net.node</i> . |

A.B.C A is an eight-bit quantity occupying the high-order byte; B is an eight-bit quantity occupying the next byte; and C is a 16-bit quantity occupying the remaining bytes. This form is suitable for a Class B address of the form **128.net.node**.

A.B.C.D The four parts each occupy a byte in the address.

Example

Engineering network

192.35.53.1	laizy.Lachman.COM laizy
192.35.53.2	laidback.Lachman.COM laidback
192.35.53.85	laiter.Lachman.COM laiter# Sun-3/50 [stevea]

Files

/etc/hosts

See Also

hostname(TC), uname(C), networks(SFF), inet(ADMP).

hosts.equiv

list of trusted hosts

Description

Hosts.equiv resides in directory */etc* and contains a list of trusted hosts. When an *rlogin*(1) or *rcmd*(1) request from such a host is made, and the initiator of the request is in */etc/passwd*, then no further validity checking is done. That is, *rlogin* does not prompt for a password, and *rsh* completes successfully. So a remote user is "equivalenced" to a local user with the same user ID when the remote user is in **hosts.equiv**.

The format of **hosts.equiv** is a list of names, as in this example:

```
host1
host2
```

A line consisting of a simple host name means that anyone logging in from that host is trusted. The **.rhosts** file has the same format as **hosts.equiv**. When user *XXX* executes *rlogin* or *rcmd*, the **.rhosts** file from *XXX*'s home directory is conceptually concatenated onto the end of **hosts.equiv** for permission checking. In the special case when the user is the super-user then only the **.rhosts** file is checked.

It is also possible to have two entries (separated by a single space) on a line of these files. In this case, if the remote host is equivalenced by the first entry, then the user named by the second entry is allowed to log in as anyone, that is, specify any name to the **-l** flag (provided that name is in the */etc/passwd* file, of course). Thus

```
laidbak ez
```

allows *ez* to log in from *laidbak* as anyone. The usual usage would be to put this entry in the **.rhosts** file in the home directory for *derek*. Then *ez* may log in as *derek* when coming from *laidbak*.

Files

```
/etc/hosts.equiv
$HOME/.rhost
```

See Also

rlogin(TC), *rcmd*(TC)

inetd.conf

configuration file for inetd (internet "super-server")

Description

inetd.conf is the configuration file for the *inetd*(SFF) System V STREAMS TCP/IP internetworking "super-server".

The file consists of a series of single-line entries, each entry corresponding to a service to be invoked by *inetd*. These services are connection-based, datagram, or "internal".

Internal services are those supported by the *inetd* program: these services are "echo", "discard", "chargen" (character generator), "daytime" (human readable time), and "time" (machine readable time, in the form of the number of seconds since midnight, January 1, 1900). All of these services are tcp based.

Each service, including internal services, must have a valid entry in */etc/services*(ADMN). In the case of an internal service, its name must correspond to the official name of the service: that is, the first entry in */etc/services*.

Each entry has a series of space- or tab-separated fields. (No field, except for the last one, may be omitted.) The fields are as follows:

service name

Name of a valid service in */etc/services*, as described above.

socket type

One of "stream", "dgram", or "raw", depending on whether the socket type is stream, datagram, or raw [see *socket* (SSC)].

protocol

Name of a valid protocol (for example, "tcp") specified in */etc/protocols*(ADMN).

wait/nowait

Specifies whether the socket can be made available for new connections while there is still data waiting on the socket. The value is always "nowait" unless it is a datagram socket. If it is a datagram socket, the value is usually "wait", although "nowait" is possible in some cases. (Note that *tftpd* is an exception in that it must have "wait" specified, and yet the socket can continue to process messages on the port.)

user

Name of the user as whom the server should run. This allows servers to be run with less permission than root.

server program

Except in the case of internal services, full pathname of the server program to be invoked by *inetd* when a request is waiting on a socket. For an internal service, the value is "internal".

server program arguments

Arguments to the server program, starting with *argv*[0], which is the name of the program. For an internal service, the value is "internal".

Comments are denoted by a "#" at the beginning of a line.

The distribution **inetd.conf** file contains prototype entries; refer to these entries when editing the file.

Example

```

.
.
.
ftp      stream tcp    nowait  root    /etc/ftpd  ftpd
telnet   stream tcp    nowait  root    /etc/telnetd telnetd
login    stream tcp    nowait  root    /etc/rlogind rlogind
exec     stream tcp    nowait  root    /etc/rexecd rexecd
finger   stream tcp    nowait  sync    /etc/fingerd fingerd
echo     stream tcp    nowait  root    internal
discard  stream tcp    nowait  root    internal
chargen  stream      tcp      nowait  rootinternal
daytime  stream      tcp      nowait  rootinternal
time     stream tcp    nowait  root    internal
echo     dgram  udp    wait    root    internal
discard  dgram  udp    wait    root    internal
chargen  dgram      udp    wait    rootinternal
daytime  dgram      udp    wait    rootinternal
time     dgram  udp    wait    root    internal
.
.
.

```

See Also

fingerd(ADMN), ftpd(ADMN), inetd(ADMN), rexecd(ADMN), rlogind(ADMN), rshd(ADMN), telnetd(ADMN), tftpd(ADMN), protocols(SFF), services(SFF).

localhosts

configuration file for sendmail

Description

Localhosts is a file that lists hosts that are to be treated as equivalent by *sendmail*(ADMN). In the distributed configuration files, an equivalent host is in class S. *Sendmail* also looks at */etc/hosts.equiv*.

The format of *localhosts* is very simple. It consists of a list of host-names, one per line. There is no support for comments.

Example

```
laidbak  
laiter  
laisagna
```

Files

/usr/lib/mail/localhosts

See ALso

hosts.equiv(SFF),
uucpindomain(SFF).

sendmail(ADMN),

sendmail(SFF),

Sendmail Installation and Operations Guide.

netrc

login file for remote networks

Description

If the **.netrc** file exists, it will be used by *ftp*(TC) for automatic login on the remote host. For each remote host, the file contains a one-line entry that describes the login data for the user on that host.

An entry may consist of up to three blank-separated fields introduced by keywords. The keyword is followed by the literal data needed for login. The following keywords are available:

machine	The hostname of the machine.
login	The user login name for that host.
password	(Optional) The user's password on that host. NOTE: The literal password must be given in clear text; it is not encrypted.

If the **.netrc** file includes the password feature, permissions on the file must be set to prohibit reading by group and others; the file will not otherwise take effect.

Example

The following example entry allows automatic login on the "admin" host by a user named "superuser" whose password is "open".

```
machine admin login superuser password open
```

Files

\$HOME/.netrc

See Also

ftp(TC).

Warning

For security reasons, use of the password feature is not recommended.

networks

names and numbers for the internet

Description

The file `/etc/networks` lists networks on the internet. Each line describes a single network and consists of the following blank separated fields:

name number aliases ...

where

name. is the official name of the network. All hosts on the internet should use the same official name for a given network.

number is the network number, which serves as part of the DARPA Internet address for each host on the internet. All hosts on the internet must use the same number for a given network.

aliases ... is a blank-separated list of local aliases for the network.

The routines which search this file ignore comments (portions of lines beginning with #) and blank lines.

Example

```
# Building 1 Internet
Lachman-Net 192.35.52    #General
LAI-TCP-Net 192.35.53    #TCP Development
```

See Also

`hosts(SFF).`

Files

`/etc/networks`

protocols

list of Internet protocols

Description

The file `/etc/protocols` lists known DARPA Internet protocols. Each line describes a single protocol and consists of the following blank separated fields:

name number aliases ...

where

name is the official name of the protocol.

number is the protocol number.

aliases ... is a blank-separated list of local aliases for the protocol.

The routines which search this file ignore comments (portions of lines beginning with #) and blank lines.

Protocol names and numbers are specified by the DDN Network Information Center. Do not change this file.

Files

`/etc/protocols`

See Also

`socket(SSC)`, `slink(ADMN)`, `ldsocket(ADMN)`.

resolver

resolver configuration file

Syntax

/etc/resolv.conf

Description

The resolver configuration file contains information that is read by the resolver routines the first time they are invoked by a process. The file is designed to be human readable and contains a list of name-value pairs that provide various types of resolver information.

On a normally configured system this file should not be necessary. The only name server to be queried will be on the local machine and the domain name is retrieved from the system.

The different configuration options are:

nameserver

followed by the Internet address (in dot notation) of a name server that the resolver should query. At least one name server should be listed. Up to MAXNS (currently 3) name servers may be listed; if more than one name server is specified, the resolver library queries each one in the order listed. If no *nameserver* entries are present, the default is to use the name server on the local machine. The algorithm used is to try a name server, and if the query times out, try the next, until out of name servers; then repeat trying all the name servers until a maximum number of retries are made.

domain

followed by an domain name, that is the default domain to append to names that do not have a dot in them. If no *domain* entries are present, the domain returned by *gethostname* (SLIB) is used (everything after the first '.'). Finally, if the host name does not contain a domain part, the root domain is assumed.

The name value pair must appear on a single line, and the keyword (e.g. *nameserver*) must start the line. The value follows the keyword, separated by white space.

Example

```
domain Lachman.COM
nameserver 192.35.52.1
nameserver 192.35.52.2
```

Files

/etc/resolv.conf

See Also

named(ADMN), resolver(SFF), hosts(ADMN), byteorder(SLIB),
rexec(SLIB).

Name Server Operations Guide for BIND

rhhosts

remote equivalent users

Description

These files grant permission for remote users to use local user names without knowing the corresponding user passwords. This is known as making the remote user "equivalent" to the local user, and is convenient, for example, when one person owns user names on more than one host.

If a user's home directory contains a file named **.rhhosts**, remote users specified in the file are equivalent to the local user. Each user specification in the file consists of the remote user host name and user name, separated by a space. (If an asterisk is substituted for either name, any name will match.) For security reasons, **.rhhosts** must belong to the user granting the equivalence or to root.

The file **/etc/hosts.equiv** is a list of remote hosts with matching-name equivalence. The file lists remote hosts one per line. On each host listed in **/etc/hosts.equiv**, a remote user with the same name as a local user is equivalent to the local user. In effect, the users are the same if the names are the same.

Files

\$HOME/.rhhosts
/etc/hosts.equiv

See Also

rcmd(TC), rcp(TC), rlogin(TC).

Warnings

When a system is listed in **/etc/hosts.equiv**, its security must be as good as local security. One insecure system mentioned in **/etc/hosts.equiv** can compromise the security of an entire network.

sendmail.cf

configuration file for sendmail

Description

Sendmail.cf is the configuration file for the sendmail mail router. A full description of this file can be found in chapter nine of the *STREAMS TCP User's Guide*.

Files

/usr/lib/sendmail.cf

See Also

sendmail(ADMN), localhosts(SFF), uucpindomain(SFF).
Sendmail Installation and Operations Guide.

services

list of Internet services

Description

The file `/etc/services` lists known DARPA Internet services. Each line describes a single service and consists of the following blank separated fields:

name number/protocol aliases ...

where:

name is the official name of the service.

number is the service number.

protocol is the name of the protocol used by the service. (See *protocols*(SFF).)

aliases ... is a blank-separated list of local aliases for the service.

The routines which search this file ignore comments (portions of lines beginning with `#`) and blank lines.

Service names and numbers are specified by the DDN Network Information Center. Do not change this file unless you are familiar with DARPA Internet internals.

Files

`/etc/services`

See Also

`inetd`(ADMN), `inetd.conf`(SFF).

/etc/sockcf

socket configuration file

Description

/etc/sockcf contains information about the protocols that are to be accessed via the socket interface. This file is read by *ldsocket* (ADMN) at boot time.

/etc/sockcf contains one line per protocol which specifies the address family, protocol type, protocol number, flags, and STREAMS device for the protocol. The flags describe the behavior of the protocol.

The format of a protocol line is:

Family	Type	Protocol	Flags	Device
--------	------	----------	-------	--------

Family can be an address family name or an integer. The following address family names are recognized:

Name	Value	Description
UNSPEC	0	Unspecified
UNIX	1	Local to host (pipes, portals)
INET	2	Internetwork: TCP, UDP, etc.
IMPLINK	3	Arpanet IMP addresses
PUP	4	PUP protocols, e.g. BSP
CHAOS	5	MIT CHAOS protocols
NS	6	XEROX NS protocols
NBS	7	NBS protocols
ECMA	8	European Computer Manufacturers
DATAKIT	9	Datakit protocols
CCITT	10	CCITT protocols: X.25, etc.
SNA	11	IBM SNA
DECnet	12	DECnet
DLI	13	Direct Data Link Interface
LAT	14	LAT
HYLINK	15	NSC Hyperchannel
APPLETALK	16	Apple Talk

Type can be a type name or an integer. The following type names are recognized:

Name	Value	Description
STREAM	1	Stream socket
DGRAM	2	Datagram socket
RAW	3	Raw protocol interface
RDM	4	Reliably delivered message
SEQPACKET	5	Sequenced packet stream

Protocol is the protocol number associated with the protocol.

Flags is a string of flag characters describing the protocol. The recognized flag characters are:

- M This protocol supports atomic messages only.
- C Connections are required.
- A Messages contain addresses.
- R Rights can be passed with this protocol.
- P The protocol number must be bound to the stream. This is required to support raw IP sockets.

/etc/sockcf may contain comments, which are delimited by '#' and newline.

The standard */etc/sockcf* contains the following entries:

INET	STREAM	6	C	/dev/inet/tcp
INET	DGRAM	17	AM	/dev/inet/udp
INET	RAW	1	AM	/dev/inet/icmp
INET	RAW	255	AMP	/dev/inet/rip

Because of the way the kernel builds the protocol switch table, the last protocol specified for a type becomes the default. For this reason, it is important to ensure that the default protocol is the last one listed.

Files

/etc/sockcf

See Also

ldsocket(ADMN), intro(ADMP), socket(SSC).

/etc/strcf

STREAMS Configuration File for STREAMS TCP/IP

Description

/etc/strcf contains the script that is executed by *slink*(SFF) to perform the STREAMS configuration operations required for STREAMS TCP/IP.

The standard **/etc/strcf** file contains several functions that perform various configuration operations, along with a sample **boot** function. Normally, only the **boot** function must be modified to customize the configuration for a given installation. In some cases, however, it may be necessary to change existing functions or add new functions.

The following functions perform basic linking operations:

Function **tp** is used to set up the link between a transport provider, such as TCP, and IP.

```
#
# tp - configure transport provider (i.e. tcp, udp, icmp)
# usage: tp devname
#
tp {
    p = open $1
    ip = open /dev/inet/ip
    link p ip
}
```

Function **linkint** links the specified streams and does a **sifname** operation with the given name.

```
#
# linkint - link interface to ip or arp
# usage: linkint top bottom ifname
#
linkint {
    x = link $1 $2
    sifname $1 x $3
}
```

Function **aplinkint** performs the same function as **linkint** for an interface that uses the **arpproc** module.

```
#
# aplinkint - like linkint, but arpproc is pushed on dev
# usage: aplinkint top bottom ifname
#
```

```

aplinkint {
    push $2 arpproc
    linkint $1 $2 $3
}

```

The following functions are used to configure different types of Ethernet interfaces:

Function **uenet** is used to configure an Ethernet interface for a cloning device driver that uses the *unit select* ioctl to select the desired interface. The interface name is constructed by concatenating the supplied prefix and the unit number.

```

#
# uenet - configure ethernet-type interface for cloning driver using
#         unit select
# usage: uenet ip-fd devname ifprefix unit
#
uenet {
    ifname = strcat $3 $4
    dev = open $2
    unitselect dev $4
    aplinkint $1 dev ifname
    dev = open $2
    unitselect dev $4
    arp = open /dev/inet/arp
    linkint arp dev ifname
}

```

Function **denet** performs the same function as **uenet**, except that *DL_ATTACH* is used instead of *unit select*.

```

#
# denet - configure ethernet-type interface for cloning driver using
#         DL_ATTACH
# usage: denet ip-fd devname ifprefix unit
#
denet {
    ifname = strcat $3 $4
    dev = open $2
    dlattach dev $4
    aplinkint $1 dev ifname
    dev = open devname
    dlattach dev $4
    arp = open /dev/inet/arp
    linkint arp dev ifname
}

```

Function **cenet** is used to configure an Ethernet interface for a cloning device driver that uses a different major number for each interface. The device name is formed by concatenating the supplied device

name prefix and the unit number. The interface name is formed in a similar manner using the interface name prefix.

```
#
# cenet - configure ethernet-type interface for cloning driver with
#       one major per interface
# usage: cenet ip-fd devprefix ifprefix unit
#
cenet {
    devname = strcat $2 $4
    ifname = strcat $3 $4
    dev = open devname
    aplinkint $1 dev ifname
    dev = open devname
    arp = open /dev/inet/arp
    linkint arp dev ifname
}
```

Function

senet

is used to configure an Ethernet interface for a non-cloning device driver. Two different device nodes must be specified for IP and ARP.

```
#
# senet - configure ethernet-type interface for non-cloning driver
# usage: senet ip-fd ipdevname arpdevname ifname
#
senet {
    dev = open $2
    aplinkint $1 dev $4
    dev = open $3
    arp = open /dev/inet/arp
    linkint arp dev $4
}
```

Function **senetc** is like **senet**, except that it allows the specification of a convergence module to be used with the ethernet driver (e.g. for the 3B2 emd driver).

```
#
# senetc - configure ethernet-type interface for non-cloning driver
#          using convergence module
# usage: senetc ip-fd convergence ipdevname arpdevname ifname
#
senetc {
    dev = open $3
    push dev $2
    aplinkint $1 dev $5
    dev = open $4
    push dev $2
    arp = open /dev/inet/arp
}
```



```

        linkint arp dev $5
    }

```

Function **loopback** is used to configure the loopback interface.

```

#
# loopback - configure loopback device
# usage: loopback ip-fd
#
loopback {
    dev = open /dev/lcloop
    linkint $1 dev lo0
}

```

Function **slip** is used to configure a SLIP interface. This function is not normally executed at boot time. Rather, the *slattach*(ADMIN) command runs *slink* specifying **slip** on the command line.

```

#
# slip - configure slip interface
# usage: slip unit
#
slip {
    ip = open /dev/inet/ip
    s = open /dev/slip
    ifname = strcat sl $1
    unitsel s $1
    linkint ip s ifname
}

```

Function **boot** is called by default when *slink* is executed. Normally, only the *interfaces* section and possibly the *queue params* section will have to be customized for a given installation. Examples are provided for the various Ethernet driver types.

```

#
# boot - boot time configuration
#
boot {
    #
    # queue params
    #
    initqp /dev/inet/udp rq 8192 40960
    initqp /dev/inet/ip muxrq 8192 40960 rq 8192 40960
    #
    # transport
    #
    tp /dev/inet/tcp
    tp /dev/inet/udp
    tp /dev/inet/icmp
    #
    # interfaces

```



```
#
ip = open /dev/inet/ip
senetc ip eli /dev/emd0 /dev/emd1 en0
#
uenet ip /dev/abc en 0
#
denet ip /dev/def en 0
#
cenet ip /dev/ghi en 0
senet ip /dev/jkl0 /dev/jkl1 en0
loopback ip
}
```

Files

/etc/strcf

See Also

slink(ADMN), intro(ADMP).

uucpindomain

configuration file for sendmail

Description

Uucpindomain is a file that lists hosts that are connected by UUCP, but should be treated as if they were in the local domain by *sendmail*(1M). In the distributed configuration files, this type of host is in class **L**.

The format of *uucpindomain* is very simple. It consists of a list of hostnames, one per line. There is no support for comments.

Example

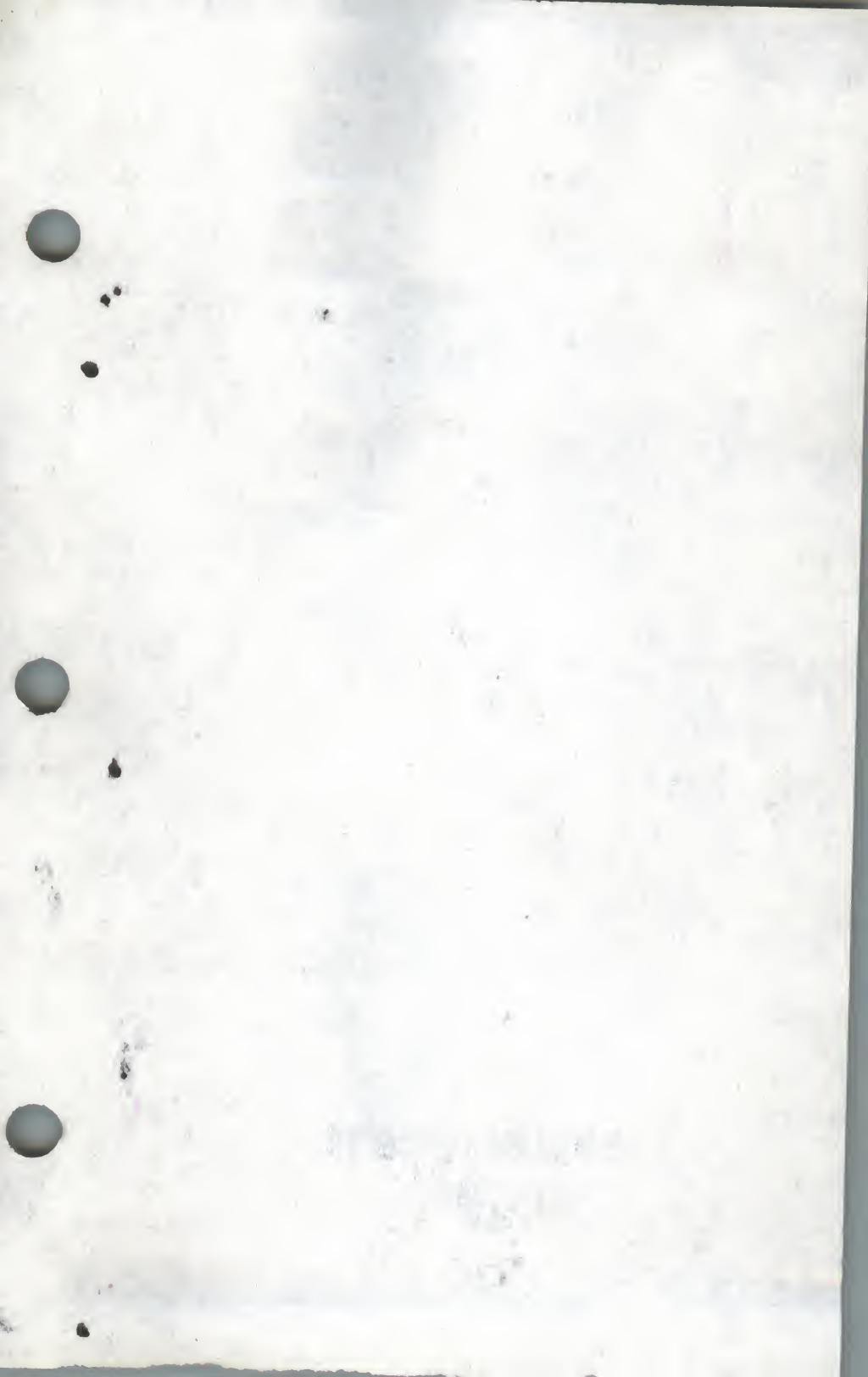
```
huey  
duey  
louie
```

Files

/usr/lib/mail/uucpindomain

See Also

hosts.equiv(SFF), localhosts(SFF), sendmail(ADMN), sendmail(SFF).
Sendmail Installation and Operations Guide.





528-210-002



54735